Aspect-Based Sentiment Analysis (ABSA) on Glassdoor Reviews

Abraham L. Apellanes
Department of Computer Science
Stanford University
aapellan@cs.stanford.edu

Andrew A. Chang
Department of Computer Science
Stanford University
achang97@cs.stanford.edu

Abstract

Glassdoor is a repository of largely unexplored text reviews that each exhibits a uniquely segmented structure. Along with its feature-level rating system, this makes Glassdoor a powerful source of data that is well-primed for exploring aspect-based sentiment analysis (ABSA). We seek to execute ABSA on this particular type of data to observe interesting patterns when it comes to the low-level, feature-specific aspects of a company. As a result, we hope to be able to more closely discern what makes a company successful in the eyes of its employees.

1 Introduction

Sentiment analysis, the task of identifying user *opinion* regarding a specific *subject*, remains one of the most popular tasks in NLU. Though identifying the sentiment toward a general topic still presents an interesting challenge, researchers have more recently turned toward the task of *aspect-based sentiment analysis*. That is, given a sentence such as, "The food was great, but the service was terrible," the task focuses not on categorizing the sentiment of the whole sentence but instead on the specific opinions toward the aspects of food and service.

Online review-based platforms like Amazon and Yelp house massive amounts of valuable information on user sentiment. With each review containing both a qualitative description and a quantitative rating of the subject, these platforms have become popular subjects for ABSA. Surprisingly, compared to the reviews on Amazon and Yelp, the data available on the website Glassdoor remains relatively unexplored. These reviews offer an arguably richer amount of detailed information on companies and their specific features. Glassdoor reviews exhibit a uniquely segmented structure and more importantly, offer aspect-specific ratings on qualities like workplace culture and employee benefits that lend themselves perfectly to the task of aspect-based sentiment analysis.

Overall, the goal of our project is to successfully execute aspect-specific sentiment analysis of companies based on their respective Glassdoor reviews. We are confident that this is achievable with a neural network classifier that leverages the unique structure of Glassdoor reviews, utilizes powerful word embeddings, and identifies phrases related to specific features. We break our general approach down into two main parts:

1. Identifying and extracting / scoring feature-relevant phrases and sentences.
2. Training multiple aspect-based sentiment classifiers that leverage the unique structure of Glassdoor reviews and feature-specific information from Part 1.

While we recognize that developing a powerful sentiment classifier is important and develop some baseline models for evaluating the success of our ABSA, we focus most of our efforts on the first stage in the pipeline. The remainder of our paper will be structured according to the following format. Section 2 discusses related work and sources of inspiration for our various approaches, and Section 3 discusses the dataset and the unique Glassdoor review structure. We discuss our models, results, and related error analysis in Sections 4, 5, and 6, before delving into a more general discussion of our task and potential future improvements in Section 7.

2 Related Work

Most of the existing literature we reviewed apply novel sentiment analysis methods to product reviews of online retailers like Amazon and Yelp, presenting powerful algorithms such as PMI-IR

(Turney et al.) and minimum cuts (Pang et al.) as alternative or additional steps to the more standard approach of training a neural network on vectorized document representations. In general, each of the individual papers focuses on different subtasks that comprise the overall goal of performing document-level or feature-level sentiment analysis.

Zhang et al and Pang et al. present insights which focus on extracting feature-relevant information. The former discuss a method for creating a feature-based product ranking system which utilizes NLP techniques and opinion analysis, constructing a directed graph to measure the relative quality of products compared to one another. They hand-define a set of relevant keywords for each aspect of interest to discover feature-relevant phrases, a technique that we will apply to our task. Alternatively, Pang and Lee's work on minimum cuts splits text into objective and subjective sentences, extracting the latter as a more concise summary of the document. Interestingly, Zhang et al. also utilize this approach of isolating solely subjective sentences, though they instead use these sentences to identify comparative relationships between product pairs. In Pang and Lee's case, feeding these summaries into their ML methods allows for discovery of cross-sentence contextual constraints and feature-specific sentiment. We leverage a similar approach in our attempt to produce feature-relevant summaries, feeding our sentiment classifier filtered versions of the original input.

Though not focused on the issue of ABSA, Gallagher et al. introduce a relevant information-theoretic topic model named CorEx (**Cor**relation **Ex**planation) which can discover the most informative topics within a dataset. They have implemented both fully unsupervised and semi-supervised versions of their algorithm, where the latter can be seeded with "anchor words" to more effectively identify distinct topics. CorEx scores represent the mutual information between the word and assigned topic, and we utilize this model to discover the feature-specific relevance of sentences and phrases.

Ding et al.'s work with holistic lexicon-based approaches to opinion mining offers a differing perspective, as they utilize lists of context-dependent opinion words and hand-defined intra-sentence and inter-sentence conjunction rules to more readily handle interactions between phrases or sentences. We choose to focus less on hand-defined semantic rules and opt to focus on models that can learn these relationships.

In contrast, Frank and Whittle utilize well-established methods on a novel dataset, leveraging GloVe embeddings with a bidirectional LSTM to predict the sentiment of company and job reviews scraped from Glassdoor. Their paper discusses the distinct structure of reviews, having already been divided into separate sections for pros and cons; as a result, the dataset demands feature extraction algorithms capable of leveraging this inherent sentiment information. We adopt their approach of utilizing GloVe embeddings and focusing on the distinct review structure, but additionally utilize the review title and advice to management sections and place considerably more focus on the identification of feature-relevant phrases.

3 Data

Our dataset contains approximately 87,000 Glassdoor reviews of 19 different companies: Airbnb, Amazon, Apple, Cisco, Deloitte, Facebook, Google, J.P. Morgan, Macy's, Microsoft, Nordstrom, Oracle, Salesforce, Square, Tesla, Uber, UPS, Visa, and Yelp.

Compared to reviews on other platforms like Amazon, Glassdoor reviews exhibit a unique structure. Product reviews on the former usually comprise three portions: (1) a title, (2) a single unstructured blob of text, and (3) an overall rating. In contrast, the main text portion of Glassdoor reviews can be broken down into three separate sections: (1) pros, (2) cons, and (3) advice to management. In addition, along with an overall rating, each Glassdoor review also provides ratings on five specific features of the company: (1) work / life balance, (2) culture & values, (3) career opportunities, (4) compensation and benefits, and (5) senior management. Each of these are ordinal variables that can be categorized as 1, 2, 3, 4, or 5-star ratings.

Overall, each entry in our Glassdoor dataset contains information on the review's:

1. date
2. company
3. employee title
4. review title
5. pros
6. cons
7. advice to management

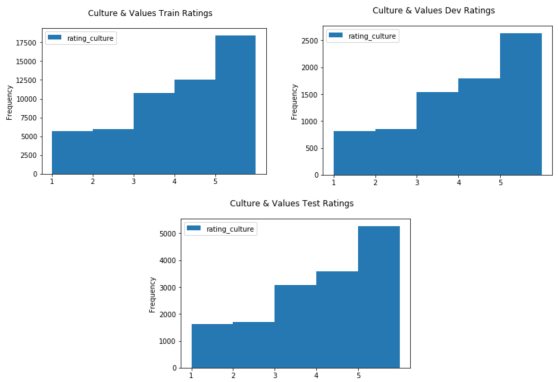

Figure 1: Stratified, left-skewed distributions of ratings for the Culture & Values feature.

8. overall rating
9. work/life balance rating
10. culture rating
11. career opportunities rating
12. compensation rating
13. senior management rating

The individual feature ratings will allow us to more easily leverage techniques for aspect-level, feature-based sentiment analysis.

The dataset is imbalanced and demonstrates a heavy left skew with many more positive than negative ratings, as demonstrated for the culture & values feature in **Figure 1**. As we obtained reviews from relatively successful companies, this distribution is consistent with our expectations.

We split the entries into separate training, development, and testing datasets, choosing to create these fixed splits instead of utilizing cross validation due to the relatively large amount of entries. As we plan to train separate classifiers for each of the five workplace features, we create distinct train / dev / test splits for each aspect in order to properly stratify the data for each of these models. **Figure 1** illustrates how the distributions of ratings for the culture values feature are approximately the same across all splits.

4 Models and Approach

As we mentioned earlier, we break down our approach into two main stages:

1. Identifying feature-relevant phrases and sentences.
2. Training multiple aspect-based sentiment classifiers.

During our implementation, we actually flipped the order of these steps. Specifically, we experimented with several different baseline classification models, before attempting to improve their performance by incorporating the feature relevance of individual sentences. Correspondingly, we will first describe the various sentiment classifiers we implemented before discussing how we calculated and utilized the feature relevance of specific phrases and sentences.

4.1 Sentiment Classifiers

Our project focuses on predicting and categorizing reviews into their correct bucket for each individual feature. That is, we will categorize each review as being a 1, 2, 3, 4, or 5-star review in relation to each of the following: (1) work / life balance, (2) culture values, (3) career opportunities, (4) compensation and benefits, and (5) senior management.

4.1.1 Bag of Words + Softmax

For our first task-specific baseline, we utilize a bag of words featurizer and a logistic regression classifier. We use nltk's word_tokenize function to split the reviews title, pros, cons, and advice to management sections and counted the frequency of each token. In addition, we leverage balanced class weights when fitting our logistic regression model to account for the skewed distribution of our data. Importantly, this model cannot distinguish between different sections and simply counts word frequencies over the entire review.

4.1.2 Structured N-grams + Softmax

Improving upon the first baseline, we modify the previous featurizer to account for the segmented structure of the review. We still utilize the word_tokenize function to split each review section into individual words, but we prepend the section label to each token before counting frequencies. That is, given a word w from section s, we represent the word as a new string s/w; for example, the word culture in the pros section would be combined to form the term pros/culture. Differentiating between the words of each section allows the logistic regression classifier to more effectively utilize information inherent in the structure, as opposed to just learning on the text contained in the overall review. We experiment with both unigrams and bigrams, using the latter in an attempt to capture more contextual information.

4.1.3 Structured GloVe + Shallow Neural Network

Our next model utilizes 300-dimensional GloVe word embeddings and a shallow, 1-ReLU neural network classifier. We represent each review as a 1200-dimensional vector, constructed by summing up the GloVe representations for each of the four review sections and then concatenating the summed vectors. Like the previous approach, this model accounts for the segmentation of each review, by allocating a different sub-vector to store the summed embeddings for each section. In addition, we move from using counts of unigrams and bigrams to word embeddings in an effort to capture more fine-grained information on context and co-occurrence.

4.2 Identifying Feature-relevant Phrases

After implementing our baseline models, we aimed to improve our model performance by identifying the relevance of individual phrases to specific features. For example, the statement "The pay is phenomenal" should contribute to a high predicted rating for the Compensation and Benefits aspect but should not impact the other four features. In our baseline models, we rely on the classifier to learn this information, but now attempt to bolster its performance by incorporating our own domain knowledge. We apply all of the following approaches to our GloVe baseline model, as we believe that the vector embeddings can capture more information than a featurizer utilizing n-grams.

As a note, for all the following approaches, we first split the reviews into separate phrases and / or sentences by first utilizing nltks sent_tokenize function. In addition, many reviews are not written with proper grammar and instead comprise of some delimiter-separated list of points. To handle this case, we split on other common delimiters if necessary (e.g. "-", ";", and ",") if sent_tokenize returns a single entry.

4.2.1 Hand-defined Keywords

As a baseline approach, we create a keyword set of around 30 words for each feature. We stem and lemmatize these keywords utilizing a Porter Stemmer and WordNet Lemmatizer respectively and filter out all phrases that do not include at least one of these keywords.

For example, some of the words in our keyword set for the Compensation and Benefits aspect include "salary," "pay," "money," "cash," and "wage"; in comparison, our keyword set for Work/Life Balance includes words like "work," "life," "balance," "time," "hours," and "PTO".

Word (Stemmed + Lemmatized)	MI Score	Log-Scale MI Score
hour	17.09	2.90
balanc	12.56	2.68
life	11.27	2.51
flexibl	11.07	2.49
day	7.15	2.25

Figure 2: Original and transformed word mutual information scores for Work/Life Balance feature.

4.2.2 WordNet Synsets

The prior approach of utilizing hand-defined synonyms is obviously extremely limited, in that it has a extremely high precision but terrible recall. In an effort to balance out these metrics, we expand the keyword sets by utilizing WordNet synsets to discover synonyms for each word, resulting in 11,000+ keywords for each category. Using these enlarged keyword sets, we apply the same filtering approach to the data in an effort to obtain feature-relevant sentences.

4.2.3 CorEx Topic Model Scores

For our optimal approach to identifying relevant phrases, we move away from our binary inclusion and exclusion based on keywords and implement a scoring mechanism to quantitatively calculate the feature relevance of each phrase. We compute this "topic relevancy score" by utilizing Gallagher et al.'s implementation of anchored CorEx, which identifies distinct topics along with their associated words and mutual information given a set of seed words. We utilize our original hand-defined keyword set as seed words and feed the CorEx model a dataset of TF-IDF vectorized sentences from our original training dataset, extracting the 500 most relevant words for each topic.

The raw mutual information scores vary greatly, so we convert them to log space by applying the transformation *log(1 + score)*. Doing so scales the scores but prevents them from becoming negative; the results of the transformation can be seen in **Figure 2**. We calculate the feature relevance for a sentence by summing the feature relevance of each word and normalizing by dividing by the length of

Model	Balance	Culture	Career	Compensation	Management
Random Classifier	0.19	0.19	0.19	0.19	0.20
Bag of Words + Softmax	0.334	0.376	0.351	0.334	0.382
Structured Bag of Words + Softmax	0.360	0.384	**0.370**	0.349	0.393
Structured Bigrams + Softmax	**0.370**	**0.387**	**0.370**	**0.367**	**0.395**
Structured GloVe + 1-ReLU NN	0.325	0.383	0.347	0.331	0.394

Figure 3: Macro-F1 scores for various classifier approaches.

the sentence. Once again, we stem and lemmatize the keywords to capture all word forms, utilizing a Porter Stemmer and WordNet Lemmatizer.

Though we could utilize the same filtering approach by defining some threshold above which we consider a sentence to be relevant, we instead feed the relevancy scores of each of the four sections (i.e. title, pros, cons, advice to management) into the model. We utilize section scores as opposed to individual sentence scores due to the variable length of reviews, as we want to feed our classifier a fixed-length vector. Computing the section score is simple and involves just summing the individual sentence topic relevancy scores. We do not normalize over the number of sentences in this case, as a section is likely to address various features and we do not want to dilute the topic relevancy scores for longer sections.

5 Results

5.1 Metric

For our metric, we will utilize the familiar quantitative classification metric of the macro-averaged F1 score. As discussed in Section 3, the dataset reveals a heavy left skew. Therefore, we choose not to simply use accuracy, as we want to observe per-class performance and properly control for size imbalance across classes. The skewed distribution of our dataset also leads us to avoid utilizing micro-averaged or weighted F1 scores, as the performance on positive reviews (4 and 5-star ratings) would dominate the final score.

5.2 Sentiment Classifiers

The results for our baseline classifiers are summarized in **Figure 3**. As expected, the models that account for the segmented Glassdoor review structure perform much better than the random classifier and the simple Bag of Words + Softmax approach. Surprisingly, the model which utilizes GloVe embeddings paired with a shallow neural network yields worse results than the combination of Structured N-grams + Softmax. Overall, the model that utilizes Structured Bigrams + Softmax yields the best results for every category.

One possible explanation for the optimal performance of this model is that bigrams capture some contextual information by examining pairs of words. In the Glassdoor reviews, there exist many bigram instances where the first word is an adjective describing the second word, which is a noun. For instance, we can examine the following example where the GloVe model fails and the bigrams model succeeds on the Work/Life Balance aspect:

> **Pros:** - Hard problems - Strong management ... - Flexible work schedule

Clearly, bigrams capture much of the important interaction between terms, such as “flexible work”.” In contrast, summing up the individual GloVe embeddings for “flexible” and “work” fails to consider contextual information and results in the misclassification of the review.

5.3 Identifying Feature-relevant Phrases

The results for our various feature relevance identification approaches are summarized in **Figure 4**. We applied all of these methods to our Structured GloVe + 1-ReLU NN model, despite the relatively poor performance of the baseline model compared to utilizing unigrams / bigrams and the logistic regression classifier. Initially, we wanted to focus on improving the sentiment classifier and utilizing

Model	Feature Relevance Identification	Balance	Culture	Career	Compensation	Management
Structured GloVe + 1-ReLU NN	N/A	0.325	**0.383**	0.347	**0.331**	0.394
Structured GloVe + 1 ReLU NN	Hard-coded keyword set	0.288	0.312	0.231	0.274	0.266
Structured GloVe + 1 ReLU NN	WordNet keyword set	0.323	0.361	0.309	0.311	0.362
Structured GloVe + 1 ReLU NN	CorEx	**0.336**	0.382	**0.358**	0.309	**0.416**

Figure 4: Macro-F1 scores for GloVe + NN classifer with various feature relevance identification approaches.

Approach	Balance	Culture	Career	Compensation	Management
Hard-coded keyword set	0.31	0.24	0.10	0.16	0.13
WordNet keyword set	0.88	0.89	0.87	0.88	0.88

Figure 5: Proportion of original sentences matched for the two filtering approaches.

Approach	1	2	3	4	5
Hard-coded keyword set	0.64	0.28	0.06	0.01	0.00
WordNet keyword set	0.02	0.01	0.02	0.11	0.83

Figure 6: Proportion of filtered sentences with 1, 2, 3, 4, and 5 relevant aspects for the two filtering approaches.

embeddings like GloVe promised an easier transition to models that could leverage contextual embeddings like ELMo or BERT. Additionally, we believed that after properly incorporating feature relevancy scores for sentences or words, the GloVe model would end up outperforming the other baseline approaches due to their more powerful, vectorized representation of words.

As we can see in **Figure 4**, utilizing both the hard-coded keyword set and the expanded WordNet keyword set failed to improve the performance of the baseline model. In fact, these approaches worsened the macro-F1 scores for each category. Understandably, the hand-defined keyword set performed the worst, as it essentially deleted large amounts of important information that could have informed the model. As we can see in **Figure 5**, the hard-coded keyword set only included 10% of the original sentences for the career aspect, leading to a huge drop-off in performance of around 11% from the initial model.

The WordNet keyword set also results in decreased performance, though it still exceeds the figures of the hard-coded approach. In this case, the issue of low recall is likely addressed, as the keyword set for each feature contains more than 11,000 words and includes at least 87% of the original sentences for each category in the filtered dataset as shown in **Figure 5**. Conversely, this approach likely has extremely low precision; in **Figure 6**, we see that over 83% of the filtered sentences in the dataset are deemed to be relevant to all five career aspects. This seems highly unlikely and probably contributes to the decreased performance of the model. Overall, these results suggest that a binary filtering approach might not be the optimal method for identification of feature-relevant phrases.

In contrast, utilizing the feature relevancy scores derived from the CorEx topic model resulted in increased performance from the Work/Life Balance, Career Opportunities, and Senior Management aspects. In particular, for the Senior Management feature, the Structured Glove + 1 ReLU NN model combined with the CorEx relevancy scores yielded the best macro-F1 score out of all explored models. In fact, the resultant score of 0.416 exceeded the next best model, which utilized bigrams and logistic regression, by over 20%. Generally, this optimal approach adds information about the feature-specific relevancy of individual sentences and review sections without removing potentially important sentences from consideration of the model.

6 Discussion

The use of NLP and NLU in the downstream tasks of sentiment analysis and classification becomes more and more important as online platforms gather greater magnitudes of reviews for products and services. However, much of the

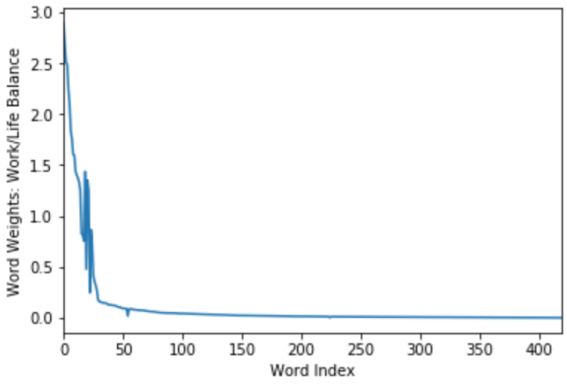

Figure 7: Relevancy scores for Work/Life Balance aspect. These are equivalent to the CorEx mutual information scores, with a *log(1 + score)* transformation applied.

well-circulated, existing SA techniques are predicated upon working with blocks of unstructured text. In this section, we further discuss the implications of our results and identify both successes and shortcomings that can potentially inform future research.

6.1 Relevancy Scoring

As discussed in our results, filtering for relevant phrases using our hand-defined keywords gave us high precision, but very low recall. The sentences that were successfully included were very often actually relevant to the topic, but many of those that were excluded also contained words that were relevant but outside of our keyword set.

Filtering using WordNet synsets produced using our initial keywords gave us the opposite problem - we saw high recall, but low precision. Blindly including all lemmas related to WordNet synsets of our seed words without filtering them on synset similarity proved to be an oversight that resulted in poor performance. It would be interesting to investigate how producing a more curated set of expanded keywords would perform on this task, especially if the word sets properly balance recall and precision. Manually curating the list or measuring similarity with WordNet or cosine distance and GloVe word embeddings could be viable first steps to accomplish this task.

After our failed attempts with sentence filtering, our goal became finding a means of incorporating relevancy in a way that proved neither too exclusive nor inclusive. Our final approach involved assigning relevancy scores to every phrase. As a result, no information would be filtered out or excluded from the sentiment classifier, but the more feature relevant phrases could still be identified by higher generated feature relevancy scores.

Figure 7 shows the scores for Work / Life Balance computed as *log(1 + score)*, where *score* is the original CorEx mutual information score associated with a word and topic. We attempted to minimize the large discrepancies we observed between highly relevant words and less relevant words. However, as **Figure 7** demonstrates, we were unable to fully convert the scores to log-space due to the fact that it would result in many scores becoming negative and difficult to interpret. In the future, it would be intriguing to see alternatives to this re-weighting scheme that maintains the non-negative properties of all scores while further alleviating the extremely small scores of less relevant words. As it stands, without the transformation applied, our relevancy scores differed by up to four or five orders of magnitude.

6.2 Incorporating Relevancy Scores Into Sentiment Classifier

Our optimal approach leveraged feature relevancy scores by computing the relevancy of individual review sections and feeding them into the classifier model. While this improved performance for several of the aspects, we believe that there are still many opportunities to more powerfully utilize these computed scores. Incorporating individual sentence feature relevance scores as opposed to section scores would be ideal, as it would capture more granular information about the review and feature-specific opinions. Combining individual feature relevancy sentence scores with sentence-level sentiment classification to produce a more general prediction for the entire review could yield interesting insights and potentially better performance.

6.3 Contextual Information

We leaned heavily on ML models to discover and learn relationships among words and text. However, other successful research has been completed on how to incorporate hand-written semantic rules (Yi et al.) and eventuated in very promising results. We believe that our feature relevancy scoring function produced through anchored CorEx offers a powerful semi-supervised method of identifying relevant feature-specific sentences. Still, it would be interesting to see how our function could leverage well-know grammatical and se-

mantic rules to even more effectively discover aspect-based sentiment.

An easier approach to capturing context could be simply using more powerful models, such as fine-tuned, pre-trained BERT models that can effectively identify and learn interactions between words. Our focus for this task shifted more to defining and computing feature relevance for words and sentences and utilizing them in sentiment classifiers, but the second portion of the pipeline stands to be improved.

7 Future Work

The amount of reviews in our dataset represents 1% of the 80M reviews available on Glassdoor. Utilizing a greater number of reviews would surely inform and generate better models. Additionally, our incorporation of feature relevancy scores into our sentiment classifier stands to be improved; instead of utilizing general section relevancy, feeding models individual sentence relevancy scores could offer heightened performance. We focused on utilizing the feature relevancy scores in the Structured GloVe + Neural Net model detailed in Section 4.1.3, but our initial baseline model testing showed that a model utilizing structured N-Grams and Softmax did better. We suggest exploring the use of synonym sets in conjunction with these models; our efforts were directed more toward the first part of the pipeline and identifying feature relevance. Lastly, as mentioned in the prior discussion section, leveraging BERT pre-trained models for multiple-choice classification and fine-tuning them on the Glassdoor data could yield promising results, capturing contextual information currently missing from our models.

Acknowledgments

We would like to acknowledge Professors Christopher Potts and Bill MacCartney, along with the CS 224U Spring teaching staff, and our project mentor, Moritz Sudhof who provided us with helpful direction throughout our project.

Authorship

Most of the responsibilities of this project were divided relatively equally between us. We co-authored the written reports for the literature review, experimental protocol, and final paper; in addition, we both worked together to create the slides and audio for our video presentation. Andrew handled the creation of the Google Cloud instance and wrote most of the initial code needed for data scraping, pre-processing, and experimentation. Abraham researched numerous papers to glean applicable approaches and ideas. Together, we discussed the performance of our models, performed qualitative and quantitative error analysis, and coded our various feature extractor and model functions.

References

[1] Ding, Xiaowen, Bing Liu, Philip Yu (2008). A Holistic Lexicon-based Approach to Opinion Mining. *Proceedings of the International Conference on Web Search and Web Data Mining*. ACM, New York, pp 231240.

[2] Frank, Fabian, Tyler Whittle (2018). Predicting Company Ratings through Glassdoor Reviews.

[3] Gallagher, Ryan J. (2017). Anchored Correlation Explanation: Topic Modeling with Minimal Domain Knowledge.

[4] Pang, Bo, Lillian Lee (2004). A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, p 271.

[5] Stoyanov, Veselin, Claire Cardie (2008). Topic Identification for Fine-Grained Opinion Analysis. Proceedings of the 22nd International Conference on Computational Linguistics, vol 1. Association for Computational Linguistics, pp 817824.

[6] Turney, Peter (2002). Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, pp 417424.

[7] Wiener, Erik, Jan Pedersen, Andreas Weigend (1995). A Neural Network Approach to Topic Spotting. Proceedings of SDAIR-95, *4th Annual Symposium on Document Analysis and Information Retrieval*. Citeseer, pp 317332.

[8] Popescu, Ana-Maria, Oren Etzioni (2005). Extracting Product Features and Opinions from Reviews. Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing.

[9] Yi, Jeonghee, Tetsuya Kasukawa, Razvan Bunescu, Wayne Niblack (2003). Sentiment Analyzer: Extracting Sentiments about a Given Topic using Natural Language Processing Techniques.

[10] Zhang, Kunpeng, Ramanathan Narayanan, Alok Choudhary (2010). Voice of the Customers: Mining Online Customer Reviews for Product Feature-based Ranking. 3rd Workshop on Online Social Networks.